

ФІЗИКО-МАТЕМАТИЧНІ МОДЕЛІ ТРАНСПОРТУ СВІТЛА ТА ЇХ АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ В АРХІТЕКТУРІ RTX

Вінницький національний технічний університет

Анотація

У доповіді розглядаються фізико-математичні засади транспорту світла у комп'ютерній графіці та сучасні методи їх обчислювальної апроксимації. Основну увагу приділено аналізу інтегрального рівняння рендерингу та його чисельному розв'язанню за допомогою методів Монте-Карло. Досліджено спеціалізовану апаратно-програмну архітектуру NVIDIA RTX, яка реалізує ці моделі через RT-ядра для апаратної обробки ієрархії обмежувальних об'ємів (BVH) та тензорні ядра для ШІ-денойзингу.

Ключові слова: транспорт світла, рівняння рендерингу, трасування променів, архітектура RTX, BVH, Монте-Карло, DLSS.

Abstract

The report considers the physical and mathematical foundations of light transport in computer graphics and modern methods of their computational approximation. The main attention is paid to the analysis of the integral rendering equation and its numerical solution using Monte Carlo methods. The specialized hardware and software architecture NVIDIA RTX, which implements these models through RT cores for hardware processing of the bounding volume hierarchy (BVH) and tensor cores for AI denoising, is investigated.

Keywords: light transport, rendering equation, ray tracing, RTX architecture, BVH, Monte Carlo, DLSS.

Вступ

Розвиток комп'ютерної графіки реального часу знаходиться на етапі фундаментального переходу від методів растеризації до фізично коректного рендерингу (Physically Based Rendering, PBR). Досягнення фотореалізму вимагає точного моделювання транспорту фотонів у тривимірному просторі, що описується складними інтегро-диференціальними рівняннями фізики оптики. Тривалий час обчислення цих рівнянь було можливим лише в офлайн-рендерингу. Поява мікроархітектури RTX (Ray Tracing Texel eXtreme) від NVIDIA запровадила спеціалізоване апаратне прискорення, яке зробило можливим наближене розв'язання фундаментальних фізичних рівнянь світла в інтерактивних додатках.

Фундаментом PBR є закон збереження енергії. Математичною формалізацією цього закону для оптичного середовища є рівняння рендерингу, вперше запропоноване Джеймсом Каджією у 1986 році [1]. Це інтегральне рівняння Фредгольма другого роду, яке описує загальну кількість світлової енергії, що випромінюється з певної точки поверхні у заданому напрямку:

$$L_o(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + \int_{\Omega} f_r(x, \omega_i, \omega_o, \lambda) L_i(x, \omega_i, \lambda, t) (\omega_i \cdot n) d\omega_i$$

де: L_o — загальна спектральна енергетична яскравість (radiance), що виходить з точки x у напрямку ω_o ; L_e — власне випромінювання поверхні в точці x ; \int_{Ω} — інтеграл по півсфері Ω вхідних напрямків над точкою x ; f_r — двонаправлена функція відбивної здатності (BRDF), що описує оптичні властивості матеріалу; L_i — світлова енергія, що надходить у точку x з напрямку ω_i ; $(\omega_i \cdot n)$ — геометричний фактор (косинус кута між вхідним променем та нормаллю n), що враховує ослаблення світла при похилому падінні (геометрична суть цих компонентів та їх взаємодія наочно подані на рис. 1).

Геометрія рівняння рендерингу

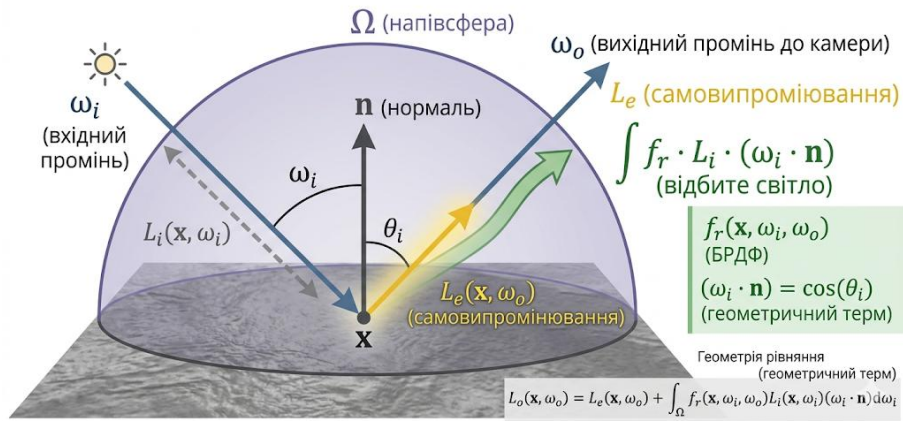


Рис. 1 – Геометрична схема транспорту світла в точці x : взаємодія вхідних (ω_i) та вихідних (ω_o) променів.

Для деталізації f_r часто використовують фізично коректні мікрогранні моделі, наприклад, модель Торренса-Спарроу, де BRDF розділяється на дифузну та дзеркальну компоненти:

$$f_r(x, \omega_i, \omega_o) = k_d \frac{c}{\pi} + k_s \frac{D(h)F(i, h)G(i, o, h)}{4(\omega_i \cdot n)(\omega_o \cdot n)}$$

де k_d, k_s — коефіцієнти дифузії та дзеркальності, D — функція розподілу мікрограней, F — коефіцієнт Френеля, G — функція геометричного затінення.

Аналітичне розв'язання рівняння Каджії для довільних сцен є неможливим. Тому використовують стохастичні чисельні методи, зокрема інтегрування Монте-Карло, яке апроксимує інтеграл як середнє значення випадкових вибірок:

$$\int_{\Omega} f(\omega_i) d\omega_i \approx \frac{1}{N} \sum_{j=1}^N \frac{f(\omega_{i,j})}{p(\omega_{i,j})}$$

де N — кількість випущених променів (вбірок), $f(\omega_{i,j})$ — значення підінтегральної функції для j -го променя, а $p(\omega_{i,j})$ — функція щільності ймовірності вибору напрямку променя. Саме цей математичний апарат лежить в основі алгоритму Path Tracing (трасування шляху), який апаратно реалізовано в RTX.

Реалізація методу Монте-Карло в реальному часі вимагає обчислення мільярдів перетинів променів з геометрією сцени на секунду. Класична архітектура GPU виконувала це на потокових мультипроцесорах (SM) програмно, що було неефективно. Мікроархітектура RTX запровадила спеціалізовані блоки — RT-ядра (Ray Tracing Cores), інтегровані в структуру потокового мультипроцесора (спрощену схему архітектури та логіку рекурсивного трасування показано на рис. 2).

СХЕМА АРХІТЕКТУРИ RTX ТА ТРАСУВАННЯ ПРОМЕНІВ

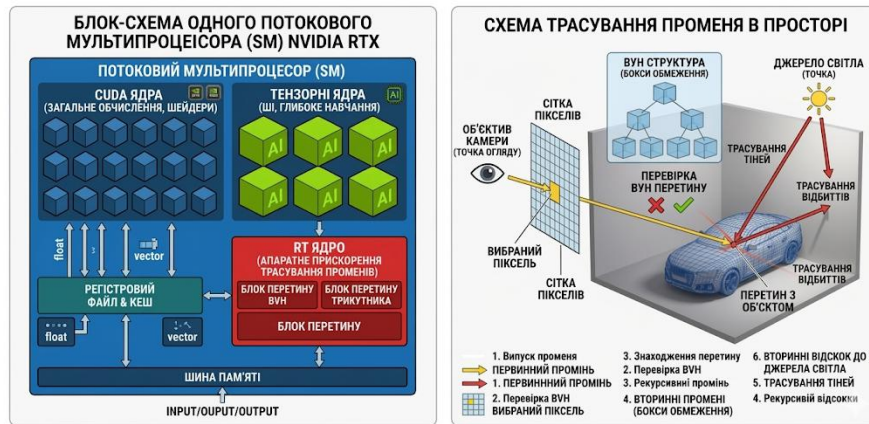


Рис. 2 – Апаратна архітектура RTX SM з RT-ядром (зліва) та логічна схема рекурсивного трасування променів (справа).

Головним завданням RT-ядер є апаратне прискорення обходу структури даних BVH (Bounding Volume Hierarchy). BVH — це деревоподібна ієрархія обмежувальних об'ємів (AABB), що дозволяє логарифмічно зменшити кількість перевірок перетину з трикутниками моделі. Принцип просторового поділу та обходу цієї структури продемонстровано на рис. 3. Програмна реалізація (API DirectX Raytracing або Vulkan) виконує наступний алгоритм:



Рис. 3 – Схема структури та обходу ієрархії обмежувальних об'ємів (BVH) RT-ядром.

RT-ядро апаратно виконує тестування «промінь-об'єм» та «промінь-трикутник», повертаючи програмі лише підтвердження перетину. Це звільняє CUDA-ядра для виконання шейдерів матеріалів. API RTX підтримує різні типи променів, як показано в таблиці 1.

Таблиця 1 – Основні типи променів в API DirectX Raytracing та їх призначення

Тип променя (Ray Type)	Фізичне призначення	Кількість вибірок (Samples per pixel)
Primary (Camera)	Первинний промінь від камери через піксель	1
Shadow	Перевірка видимості джерела світла з точки (тіні)	1-2
Reflection/Refraction	Відбиття від дзеркальних або заломлення у прозорих поверхнях	1-4+ (рекурсивно)
Indirect Radiance	Збір розсіяного світла для Глобального Освітлення (GI)	1-N (стохастично)

Оскільки розрахунок рівняння Каджії методом Монте-Карло в реальному часі дозволяє випустити лише 1-2 промені на піксель, результат містить значний високочастотний шум. Для розв'язання цієї проблеми в архітектурі RTX використовуються тензорні ядра (Tensor Cores), які виконують матричні множення з напівточністю для роботи нейронних мереж. Загальну схему інтеграції цих ШІ-алгоритмів у процес рендерингу наведено на рис. 4.

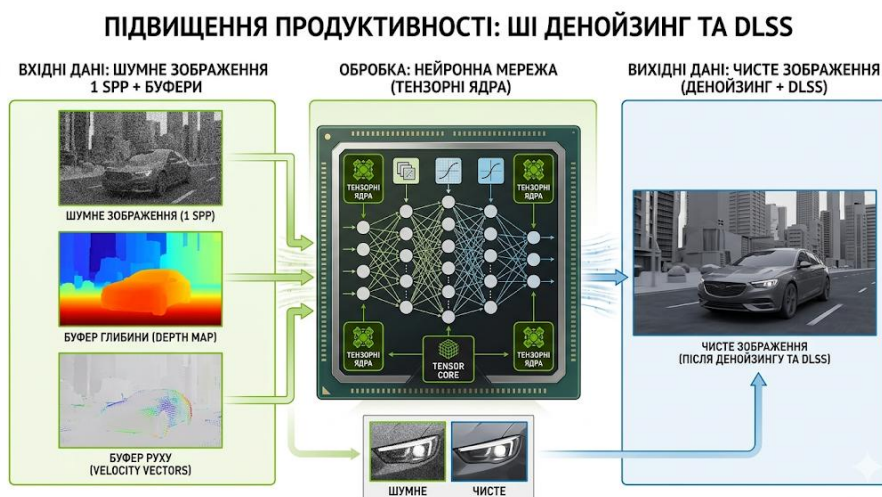


Рис. 4 – Схема інтеграції ШІ-алгоритмів (Денойзинг та DLSS) у процес рендерингу RTX

Оскільки розрахунок рівняння Каджії методом Монте-Карло в реальному часі дозволяє випустити лише 1-2 промені на піксель, результат містить значний високочастотний шум. Для розв'язання цієї проблеми в архітектурі RTX використовуються тензорні ядра (Tensor Cores), які виконують матричні множення з напівточністю для роботи нейронних мереж.

Разом із технологією DLSS (Deep Learning Super Sampling), це дозволяє рендерити сцену у меншій внутрішній роздільній здатності зі значним підвищенням продуктивності (Таблиця 2).

Таблиця 2 – Порівняльний аналіз швидкості рендерингу та фізичної точності

Метод рендерингу	Внутрішня роздільна здатність	Обчислення фізики світла	Рівень шуму (до фільтрації)	Відносний FPS (RTX 4090)
Растеризація (Native)	4K	Низька (локальні наближення)	Відсутній	100%
Path Tracing (Native, без RT яд.)	4K	Повне рівняння Каджії	Екстремальний (1 spp)	~3-5%
Path Tracing (Native, з RT яд.)	4K	Повне рівняння Каджії	Високий (1 spp)	~25-30%
Path Tracing + DLSS 3 + RT	1080p (Upscaled)	Повне рівняння Каджії + ШІ реконструкція	Компенсовано Денойзером	~120-150%

Висновки

Апаратно-програмна реалізація методів розв'язання інтегрального рівняння рендерингу в архітектурі RTX є прикладом успішної синергії фізики оптики, прикладної математики та мікроелектроніки. Перекладання обчислювально важких задач ініціалізації транспорту променів (обхід BVH) на спеціалізовані RT-ядра, а задач мінімізації дисперсії (шуму) на тензорні ядра дозволило вперше наблизити інтерактивну комп'ютерну графіку до кінематографічного рівня фізичної точності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kajiya J. T. The rendering equation // ACM SIGGRAPH Computer Graphics. – 1986. – Vol. 20, No. 4. – P. 143-150.
2. Pharr M., Jakob W., Humphreys G. Physically Based Rendering: From Theory to Implementation. – 3rd ed. – Morgan Kaufmann, 2016. – 1266 p.
3. Haines E., Akenine-Möller T. Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs. – Apress, 2019. – 642 p.
4. Shirley P., Morley R. K. Realistic Ray Tracing. – 2nd ed. – A K Peters/CRC Press, 2003. – 256 p.
5. NVIDIA RTX™ Technology [Electronic resource]. – Mode of access: <https://www.nvidia.com/en-us/geforce/rtx/>.

Коваль Олександр Дмитрович - студент групи 2ПП-25Б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: oleksandr.kov.dm@gmail.com

Науковий керівник: **Мартинюк Володимир Валерійович** — канд. техн. наук, доцент кафедри загальної фізики, Вінницький національний технічний університет, м. Вінниця

Koval Oleksandr Dmytrovych – Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksandr.kov.dm@gmail.com

Scientific supervisor: **Volodymyr Valeriiovych Martyniuk** — Cand. Sc. (Eng), Assistant Professor of general Physics, Vinnytsia National Technical University, Vinnytsia